

DOGF를 이용한 분산 프로그램 개발 Tool-Kit의 구축

임정택*, 안동인*, 신창선**, 주수종*
*원광대학교 전기·전자 및 정보공학부
**순천대학교 정보통신공학부

e-mail: *{jtlim, ahndong, scjoo}@wonkwang.ac.kr
**csshin@sunchon.ac.kr

The Construction of Distributed Program Developing Tool-Kit using DOGF

Jeong-Taek Lim*, Dong-In Ahn*, Chang-Sun Shin**, Su-Chong Joo*

*School of Electrical, Electronic and Information Engineering,
Wonkwang University

**School of Information and Communication Engineering,
Sunchon National University

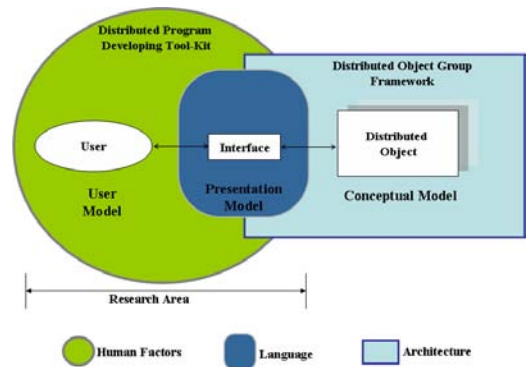
요 약

분산응용을 개발하는데 효율적인 환경을 제공해주는 분산 객체그룹 프레임워크(Distributed Object Group Framework)를 이용한 분산 프로그램 개발 Tool-Kit을 구축하였다. 본 Tool-Kit은 분산응용 개발자들에게 DOGF의 기능을 쉽게 이용할 수 있도록 지원하기 위해 객체그룹 운영자 GUI, 서버 프로그램 개발자 GUI, 클라이언트 프로그램 개발자 GUI 등 3가지 GUI로 구축하였다. 분산응용의 개발 시, 서버 프로그램 개발자는 서버 객체들 구현한 후 프레임워크에 등록 및 클라이언트에 대한 접근권한을 부여하고, 클라이언트 프로그램 개발자는 자신에게 접근권한이 부여된 이들 객체 또는 객체그룹을 검색하여 클라이언트 프로그램을 작성할 수 있도록 지원한다. 본 논문에서는 분산응용 개발 시 이용되는 Tool-Kit과 DOGF의 상호동작을 정의하고, 분산 프로그램 개발 Tool-Kit을 구축하고, 간단한 분산응용을 이용해 Tool-Kit의 수행성을 보인다.

1. 서론

현대 사회에 들어서 인터넷이 보편화되고 컴퓨터의 성능이 향상되었으며, 광역 네트워크 환경의 구축으로 단일시스템에서 개발되던 응용들이 분산시스템 상에서 복잡한 상호작용을 통해 서비스를 수행하도록 개발되고 있다. 이러한 분산시스템 환경에서 개발되는 서비스는 단일시스템과 달리 네트워크와 시스템 자원의 공유, 부하문제를 추가적으로 고려해야만 한다[1,2,3]. 즉, 분산응용이 수행 될 때, 서로 다른 시스템의 구성요소간의 복잡한 상호동작을 최소화 할 필요가 있다. 이를 위해 분산환경에서 클라이언트의 원격서비스 요청 시, 분산 프레임워크 상에서 분산응용의 수행을 향상시키기 위한 분산자원 관리 기술들이 필요하다[4,5].

우리는 위에서 언급한 요구사항들을 만족하기 위해, 분산환경에서 분산응용 서비스를 수행하는 객체들의 그룹관리 및 분산서비스를 제공하는 분산 객체그룹 프레임워크(Distributed Object Group Framework, DOGF)를 연구했었다[6,7].



본 논문에서는 분산응용 개발 시 이용되는 Tool-Kit과 DOGF의 구성요간의 상호동작을 정의하고, 분산응용 개발자들이 DOGF가 제공하는 기능들을 쉽게 이용할 수 있도록 분산 프로그램 개발 Tool-Kit을 3가지의 GUI로 구축하고, 간단한 분산응용을 이용해 Tool-Kit의 수행성을 확인해본다.

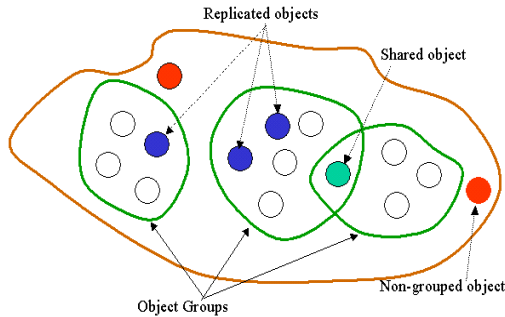
2. 관련연구

2.1 객체그룹

분산응용을 구성하는 각각의 객체들은 분산된 네

* 본 연구는 헬스케어기술개발센터를 통해 과학기술부로부터 지원을 받았음.

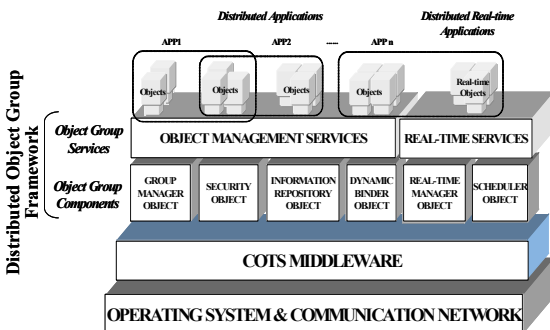
트위크 환경에서 단일 또는 중복객체들로 존재할 수 있다. 또한 다른 분산응용에서 사용된 분산객체들은 다른 응용에서 재사용될 수 있다. 각 서버 시스템에 존재하는 분산객체들은 다수의 분산응용들의 구현을 위해 효율적인 그룹 관리 및 공유를 통해서 분산 응용에 적절한 서비스를 제공할 수 있어야 한다. 이를 위해, 분산객체그룹 환경에서는 하나의 분산응용을 구성하는 분산객체들은 하나의 논리적인 그룹으로 관리되며, 클라이언트는 객체그룹 내의 분산객체들에 대한 식별자나 중복 상황에 대한 인지 없이도 서비스를 요청할 수 있어야 한다. 즉, 분산 시스템 상에 수행되는 하나의 분산응용은 하나 또는 그 이상의 객체그룹들로 구성될 수 있으며, 이들은 하나의 분산 응용의 수행을 위해 논리적으로는 단일 뷰 시스템으로 관리될 필요가 있다. 아래 (그림 2)는 분산 환경에 존재하는 객체들을 그룹핑하여 다양한 분산 응용으로 구성하는 객체그룹의 영역을 보인다.



(그림 2) 분산응용에 대한 객체그룹들의 도메인

2.2 분산 객체그룹 프레임워크

DOGF는 통신 및 미들웨어 계층과 분산응용 계층의 사이에 존재하며, 크게 객체그룹관리 지원 컴포넌트와 실시간 서비스 지원 컴포넌트로 구성된다. 분산 객체그룹 프레임워크의 객체그룹관리 지원 컴포넌트로 그룹관리자객체(Group Manager object), 보안객체(Security object), 정보저장소객체(Information Repository object), 동적바인더객체(Dynamic Binder object)로 구성되며, 실시간 서비스 지원 컴포넌트로 실시간관리자객체(Real-Time Manager object)들과 스케줄러객체(Scheduler object)로 구성된다. 우리는 분산응용의 한 예로 [7]에서 DOGF를 분산 방어시스템에 적용하여 지원 서비스들의 수행성을 확인했다. 아래 (그림 3)은 DOGF의 구조를 보인다.



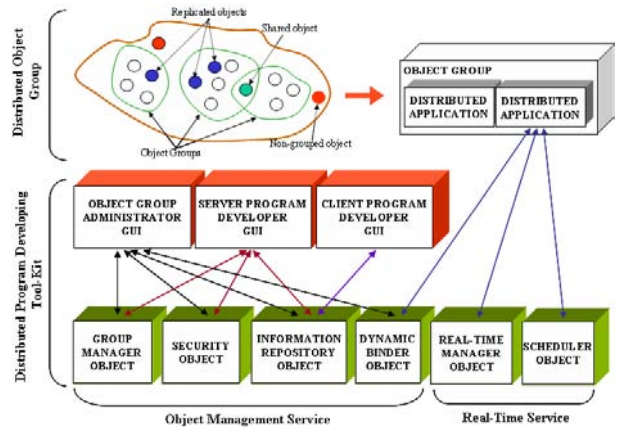
(그림 3) 분산 객체그룹 프레임워크 구조

3. 분산응용과 Tool-Kit, DOGF의 상호동작

객체그룹 운영자 GUI는 DOGF의 그룹관리자객체를 이용하여 객체그룹의 등록 및 철회, 그룹의 재구성을 할 수 있다. DOGF의 보안객체를 이용하여 객체그룹 또는 객체에 대한 접근권한 설정할 수 있으며, DOGF의 정보저장소를 통해 객체의 속성정보를 등록 및 철회, 저장된 객체의 속성 정보를 수정할 수 있다. DOGF의 동적바인더 객체를 이용하여 동적바인딩 알고리즘 선택하는 인터페이스를 제공한다.

서버 프로그램 개발자 GUI는 DOGF의 그룹관리자객체를 이용하여 자신이 개발한 서버 객체를 기존 그룹에 등록하거나 새로운 그룹을 만들어 등록할 수 있다. 자신이 등록한 서버 객체에 한해서 삭제도 가능하다. DOGF의 보안객체를 이용하여 자신이 등록한 서버 객체에 한해서 접근권한을 설정할 수 있다. DOGF의 정보저장소를 통해 자신이 개발한 서버 객체의 속성정보를 등록 및 철회, 자신이 등록한 서버 객체의 속성정보를 수정할 수 있는 인터페이스를 제공한다.

클라이언트 프로그램 개발자 GUI는 DOGF의 정보저장소를 통해 저장된 객체그룹과 서비스들 중 이용 가능한 서비스를 검색한다.



(그림 4) 분산응용과 Tool-Kit, DOGF의 상호동작

분산응용들은 물리적으로 분산되어있지만 DOGF에 의해 논리적인 그룹들로 관리된다. 중복된 서비스를 제공하는 분산응용들은 동적바인더객체에 의해 클라이언트 요청들의 동적바인딩이 이루어진다. 클라이언트의 요청을 받은 분산응용들은 실시간 서비스를 위해서 DOGF의 실시간 서비스를 이용하여 클라이언트들의 스케줄링을 요청, 실시간으로 클라이언트들 요청을 처리한다.

4. 분산 프로그램 개발 Tool-Kit의 구축

분산응용을 효율적인 객체그룹 단위로 관리하고, 서버 및 클라이언트 개발자들이 DOGF의 기능들을 편리하게 이용할 수 있도록 객체그룹 운영자 GUI, 서버 프로그램 개발자 GUI, 클라이언트 프로그램 개발자 GUI등 3개의 GUI로 구축했다.

객체그룹 운영자 GUI는 분산응용의 수행환경 설정 및 서버객체의 관리를 책임지고 운영한다. 객체그룹 운영자 GUI에서는 객체그룹 내의 중복객체들 중 하나의 객체를 선정하기 위한 기법으로 부하균형화를 고려한 바인딩 알고리즘을 선택할 수 있다. 또한 분산응용을 구성하는 서버객체들을 그룹에 등록 및 철회할 수 있으며, 서비스들에 대한 클라이언트 접근권한을 설정할 수 있다.

서버 프로그램 개발자 GUI에서는 서버 프로그램 개발자 자신들이 구현한 서버 객체들의 그룹 등록 및 철회, 접근권한을 설정할 수 있다. 객체그룹 운영자 GUI에서와 달리 서버 프로그램 개발자 GUI에서는 자신들이 구현한 서버 객체에 한해서 그룹설정 및 권한설정을 할 수 있다.

클라이언트 프로그램 개발자 GUI에서는 분산응용에 서비스 요청 프로그램을 개발하기 위해, 사용할 수 있는 그룹과 서비스들을 검색할 수 있는 기능을 제공한다. 객체그룹에 등록된 서버 프로그램의 속성정보를 검색하면 검색결과로 해당 클라이언트가 사용할 수 있는 객체그룹과 서비스명이 출력된다.

Tool-Kit에서 DOGF의 객체그룹관리서비스를 이용하기 위해 이용하는 API 호출 형식은 (그림 5)와 같다.

```

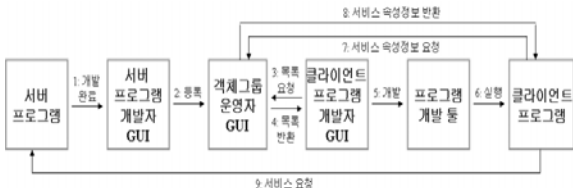
/* 객체그룹관리서비스 메소드 형식 */
char* GM(int number, char* parameter1,
char* parameter2, ... );
    
```

(그림 5) DOGF의 객체그룹관리서비스 메소드 호출

매개변수 number는 제공되는 서비스들 중 어떤 서비스를 받을 것인지 선택한다. 0은 서비스 검색, 1,2는 접근권한 설정과 삭제에 사용되며, 3,4,5는 객체그룹 생성, 삭제와 객체들의 속성정보를 등록, 삭제, 수정한다. 6은 다양한 동적바인딩 알고리즘을 설정하는데 사용된다.

5. 분산 프로그램 개발 Tool-Kit 수행

서버 개발자는 서버 프로그램을 개발한 후 서버 프로그램 개발자 GUI를 이용해 객체그룹 운영자 GUI에서 등록한다. 클라이언트 개발자는 자신이 이용할 서비스 목록을 보기 위해 클라이언트 프로그램 개발자 GUI를 이용하여 객체그룹 운영자 GUI에게 서비스 목록을 요청한다.



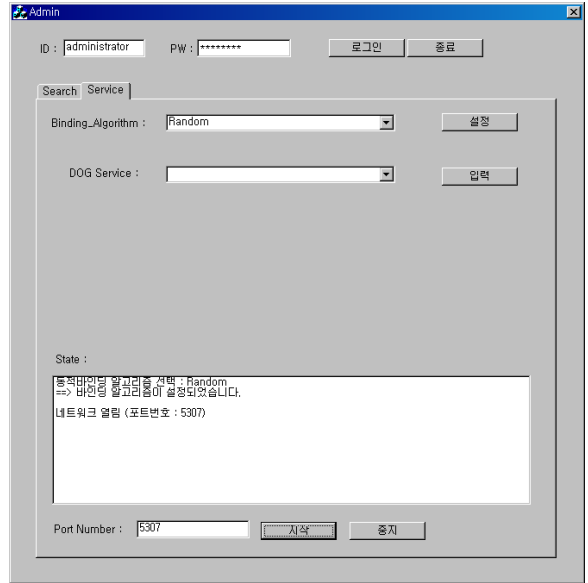
(그림 6) 분산 프로그램 개발 Tool-Kit 수행 절차

클라이언트 개발자는 자신이 이용할 수 있는 서버 프로그램 목록을 확인 후 프로그램 개발 툴을 이용하여 클라이언트 프로그램을 개발한다. 클라이언트

프로그램은 객체그룹 운영자 GUI에게 자신이 이용할 서비스의 속성정보를 요청하여 응답받은 후 서버 프로그램에 접속하여 서비스를 받는다. 위 (그림 6)은 이에 대한 수행 절차를 보이고 있다.

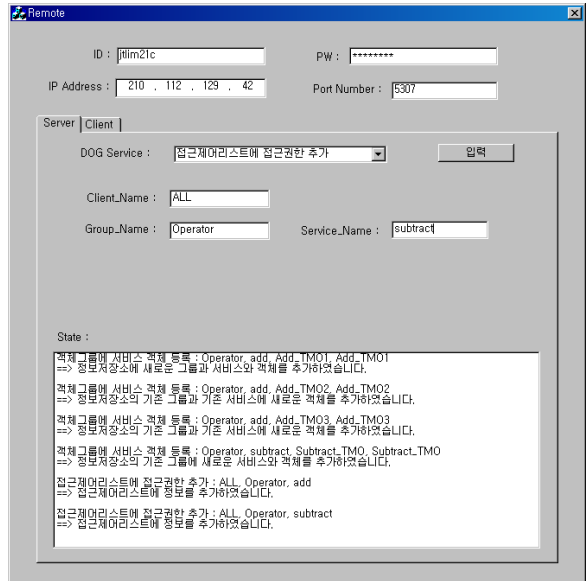
앞서 언급한 수행 절차에 따른 수행 과정을 개발 Tool Kit이 제공하는 GUI로 다음과 같이 보인다.

객체그룹 운영자는 각 서버 개발자와 클라이언트 개발자, 클라이언트 프로그램으로부터 요청을 받아 처리하고, 동일한 서비스를 제공하는 중복객체에 대한 동적 바인딩 알고리즘을 선택할 수 있다. 간단한 분산응용을 동작시키기 위해 동적바인딩 알고리즘으로 Random을 선택하였고, 포트 5307번을 열어 서버 개발자, 클라이언트 개발자, 클라이언트 프로그램으로부터 요청을 받을 수 있다.



(그림 7) 객체그룹 운영자 GUI

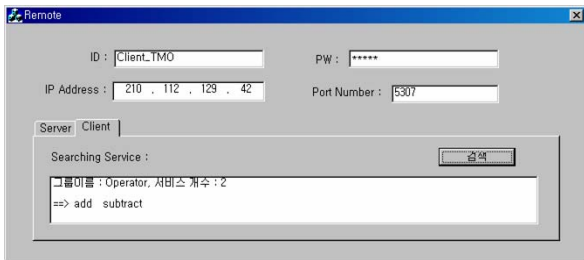
서버 개발자들은 각 서버시스템에서 서비스를 제공할 수 있는 서버응용들을 개발하고 등록해야 한다.



(그림 8) 서버 프로그램 개발자 GUI

서버 프로그램 개발자는 add, subtract 서비스를 개발한 후 DOGF에 Operator라는 객체그룹으로 add, subtract 서비스를 등록하였다. add 서비스를 제공하는 객체는 총 3개(Add_TMO1, Add_TMO2, Add_TMO3)로 중복되어 있다. add, subtract 서비스에 대한 클라이언트 접근권한은 ALL로 부여함으로써 모든 클라이언트들이 이 서비스를 이용할 수 있다.

클라이언트 개발자들은 등록된 서비스들을 이용할 수 있는 클라이언트 프로그램을 개발하기 위해 클라이언트 프로그램 개발자 GUI를 이용하여 제공되는 서비스들을 검색한다. 자신이 접근할 수 있는 검색된 서비스들을 확인하고, 서비스를 이용할 수 있는 프로그램을 개발한다.



(그림 8) 클라이언트 프로그램 개발자 GUI

Client_TMO가 접근할 수 있는 서비스들은 Operator 그룹의 add, subtract 서비스이다.

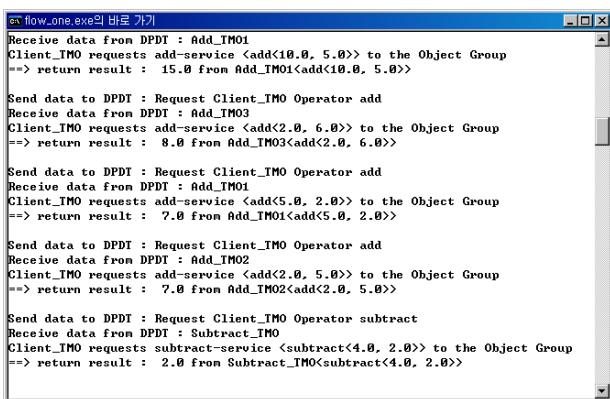
클라이언트 프로그램은 객체그룹 운영자 GUI에게 소켓으로 문자열 형태의 데이터를 전송한다. (그림 9)에서는 Client_TMO가 Operator 그룹의 add 서비스를 요청하였다.

```

/* 자신이 이용할 서비스의 레퍼런스 요청 */
{Request 클라이언트이름 그룹이름 서비스이름}

char ptr[100] = {"Request Client_TMO Operator add"}
send(Socket, ptr, strlen(ptr), 0);
    
```

(그림 9) 클라이언트 프로그램의 서비스 요청



(그림 10) 클라이언트 프로그램의 수행결과

(그림 10)의 결과를 보면 add 서비스를 4번 요청하고, subtract 서비스를 1번 요청하였다. add 서비스를 제공하는 객체는 중복 객체이므로 동적 바인딩

객체를 이용하여 분산되게 요청하였다. 위 결과의 동적바인딩 알고리즘은 Random을 이용하였다.

6. 결론 및 향후연구

본 논문에서는 분산응용 개발 시 이용되는 Tool-Kit의 구성요소와 분산응용간의 상호동작을 정의하고, DOGF를 이용한 분산 프로그램 개발 Tool-Kit을 구축하였다. 그리고 이를 간단한 분산응용을 이용해 Tool-Kit의 수행성을 확인했다. 분산응용 개발자들이 DOGF의 기능들을 편리하게 지원할 수 있도록 객체그룹 운영자 GUI, 서버 프로그램 개발자 GUI, 클라이언트 프로그램 개발자 GUI로 구축하였고, 동작절차를 살펴보았다.

향후 연구로, 본 Tool-Kit을 사용자 편의성을 제고한 기능을 추가하고, 다양한 분산 응용에 적용할 예정이다. 특히, 유비쿼터스 환경에서 요구되고 있는 헬스케어 서비스 지원을 위한 응용들에 적용하여 헬스케어기기, 장비 및 바이오센서 그리고 의료정보시스템들을 총괄적으로 통합 및 관리서비스를 지원할 수 있도록 확장에 관한 연구를 하고자 한다.

참고문헌

- [1] Bellissard, L., Boyer, F., Riveill, M., and Vion-Dury, J.-Y., "System Services for Distributed Application Configuration", In Proceedings of the 4th International Conference on Configurable Distributed Systems, 1998, pp.53-60
- [2] M. Takemoto, "Fault-Tolerant Object on Network-wide Distributed Object-Oriented Systems for Future Telecommunications Applications", In IEEE PRFTS, 1997, pp.139-146.
- [3] P.M. Melliar-Smith, L.E. Moser, and P. Narasimhan, "Consistent object replication in the Eternal System", Theory and Practice of Object System, Vol.4, No.2, 1998, pp.81-92.
- [4] V. Kalogeraki, P.M. Melliar-Smith, and L.E. Moser, "Dynamic Scheduling for Soft Real-Time Distributed Object Systems", In Proceedings of the IEEE 3rd International Symposium on Object-Oriented Real-Time Distributed Computing, 2000, pp.114-121.
- [5] G. Cooper, L. DiPippo, L. Esibov, R. Ginis, R. Johnston, P. Kortman, P. Krupp, J. Mauer, M. Squadrito, B. Thuraisingham, S. Wohlever, and V. Wolfe, "Real-Time CORBA Development at MITRE, NRaD, Tri-Pacific and URI", In Proceedings of IEEE Workshop on Middleware for Distributed Real-time Systems and Services, 1997, pp.69-74.
- [6] C.S. Shin, M.S. Kang, C.W. Jeong, and S.C. Joo, "TMO-Based Object Group Framework for Supporting Distributed Object Management and Real-Time Services", Lecture Notes in Computer Science, Vol.2834, pp.525-535, 2003.
- [7] Chang-Sun Shin, Chang-Won Jeong, and Su-Chong Joo, "Construction of Distributed Object Group Framework and Its Execution Analysis Using Distributed Application Simulation", Lecture Notes in Computer Science, Vol. 3207, pp. 724-733, 2004. 7.